Práctica #3: Uso de Clases y Métodos

Introducción

El alumno creará programas en java que manejen clases y métodos. Para hacer esto se debe recordar lo visto con respecto al manejo de clases. Un ejemplo de cómo se utilizan las clases en java es el siguiente:

```
//Ejemplo de Características de una playera
//(nombre alumno)
public class Ropa{
    public static void main(String args[]) {
      Playera miPlayera = new Playera();//declara la instancia al objeto clase
      miPlayera.asignavalores();//invocamos al método de la clase
      miPlayera.imprimeDatos();//invocamos al método de la clase
    }
}

class Playera{
      int playeraID, cantidad;
      String descripcion, color;
      double precio;

      public void asignavalores(){
             playeraID = 1;
             descripcion = "Playera tipo polo";
             color = "Rojo";
             precio = 249.99;
             cantidad = 5;
      }

      public void imprimeDatos(){
             System.out.println("ID: " + playeraID);
             System.out.println("Descripcion Playera: " + descripcion);
             System.out.println("Color: " + color);
             System.out.println("Precio: " + precio);
             System.out.println("Cantidad en Inventario: " + cantidad);
      }
};
```

Fundamento Teórico

Declaración de una clase

Es recomendable crear un archivo de texto en el block de notas, en donde declararemos una clase por cada objeto identificado en nuestro problema, aunque es posible declarar varios

objetos dentro de un mismo archivo de texto. La sintaxis para declarar una clase es la siguiente:

```
[modificadores] class nombreclase{
}
```

Si declaramos más clases en el mismo documento en donde se encuentra nuestra clase principal, la sintaxis es la siguiente:

```
class nombreclase{
};
```

En donde:

- **[modificadores]** determinan la accesibilidad de otras clases a la clase que se está declarando, así como el comportamiento de su herencia. Todos son opcionales y por el momento se empleará public.
- La palabra reservada **class** es obligatoria y le dice al compilador que el bloque de código siguiente es una declaración de clase.
- **nombreclase** será el nombre de la clase según las reglas vistas con anterioridad. Como se mencionó ya, el nombre de la clase debe coincidir exactamente con el nombre del archivo .java que la contiene.
- **Las llaves de apertura y cierre ({ })** delimitan el cuerpo de nuestra definición de clase. Todas las variables y métodos de la clase deberán ir encerrados en este bloque.

Declaración de los métodos de una clase

Los métodos siguen a los atributos de la clase; su sintaxis es la siguiente:

```
[modificadores] tipoderetorno identificador([argumentos]){
      bloque_de_instrucciones
}
```

En donde:

- **[modificadores]** representan diversas palabras reservadas que definen la forma en que los métodos son accedidos. Todos son opcionales y por el momento se empleará public.
- **tipoderetorno** indica el tipo de valor que el método regresa (en caso de haberlo). Los valores retornados por un método pueden ser empleados por el método que los invoca y

podrá ser a lo mucho uno. Si el método no retorna nada, la palabra clave void deberá usarse como tipo de retorno.

- **identificador** será el nombre del método según las reglas vistas con anterioridad.
- **[argumentos]** representa la lista de variables cuyos valores son pasados al método para que éste trabaje y son opcionales. Los paréntesis de apertura "(" y cierre ") " son obligatorios aún cuando el método no requiera ningún argumento.
- **bloque_de_instrucciones** será la lista de sentencias que el método ejecuta. Una gran variedad de tareas pueden realizarse en el bloque de código o cuerpo de un método. Así como en otros lenguajes de programación, podrán crearse tantos bloques de instrucciones internos como se desee o convenga.
- **Las llaves de apertura y cierre ({ })** delimitan el cuerpo de nuestro método.

La sintaxis genérica es la siguiente:

```
nombredeclase identificador = new nombredeclase();
```

En donde:

- **nombredeclase** representa el nombre de la clase de objeto a instanciar.
- identificador será el nombre que se le asignará a la instancia (también llamada referencia de objeto).
- **new** será la palabra clave encargada de instanciar un objeto empleando como plantilla a la clase especificada por nombredeclase.

Lo anterior puede realizarse también en dos líneas:

```
nombredeclase identificador;
identificador = new nombredeclase();
```

Por último, invocamos el método del objeto empleando una sintaxis genérica:

```
identificadordeobjeto.identificadordemétodo();
```

Dado que la sintaxis completa que se emplea para invocar un método varía considerablemente según la situación, se verá con detalle en un apartado más adelante.

Java no permite el uso de apuntadores por parte del programador, lo cual podría verse como una desventaja aunque en realidad es todo lo contrario, gracias a que no puede haber manipulación de memoria es imposible invadir entornos ajenos al de nuestras aplicaciones y por lo tanto la seguridad de nuestro sistema estará siempre garantizada.

Objetivos de Aprendizaje

El propósito de esta práctica es que el estudiante se familiarice con el entorno de la programación en Java, mediante el uso de comandos básicos al utilizar Clases y métodos.

Descripción de la práctica

1. Crear dentro de la carpeta "**Programación**" otra carpeta con el nombre de "**Practica3**".
2. Realizar el código fuente de "**Elevador**", utilizando el block de notas y guardar el archivo como **.java** en la carpeta de "**Practica3**".
3. Compilar el archivo mediante la línea de comandos, en caso que el compilador marque un error, corregir el código fuente, y compilar nuevamente. Si no hay error en el código fuente, ejecutar el programa.
4. Abrir un nuevo documento en el block de notas y realizar el programa de "**hola mundo**" utilizando clases, guardar el archivo como **.java** en la carpeta de "**Practica3**".
5. Compilar el archivo mediante la línea de comandos, en caso que el compilador marque un error, corregir el código fuente, y compilar nuevamente. Si no hay error en el código fuente, ejecutar el programa.
6. Abrir un nuevo documento en el block de notas y realizar un programa llamado "**Catalogo**", el cual, contendrá la clase Ropa, que contenga 2 tipos de prendas (Playera, Pantalón, Short, Vestidos, Sacos, etc.) y las características de las prendas (Identificador, descripción, color, cantidad en inventario y precio), para esto, crear un Array que almacene 4 prendas.
7. Compilar el archivo mediante la línea de comandos, en caso que el compilador marque un error, corregir el código fuente, y compilar nuevamente. Si no hay error en el código fuente, ejecutar el programa.
8. Realizar un reporte con los resultados obtenidos. El reporte debe contener Número de práctica, nombre de la experiencia educativa, nombre del alumno, matrícula, fecha de elaboración, desarrollo de la práctica, y resultados.
9. Compilen y ejecuten el siguiente código para ver otros ejercicios:

```
public class Ejercicios3 {
   public static void main(String[] args){
       System.out.println(“ejercicios Practica 3");
       System.out.println(“elaborar utilizando clases");

       System.out.println("1.- Crear una calculadora con los métodos de suma, resta, 
multiplicación y división");

       System.out.println("2.- realizar en un método la operación x = num1 * num2, y 
el resultado de esta operacion lo enviaran a otro método el cual diga si el resultado 
es un numero par o impar");

       System.out.println("3.- realizar la conversión entre grados fahrenheit y 
celsius");
   }
}
```

Materiales a emplear en la práctica

- Computadora con el JDK instalado.
- Block de notas
- Línea de comandos o terminal.

Bibliografía

Froute Quintas, A. (2008). JAVA 2: Manual de usuario y tutorial (5th ed.). México: Alfaomega, ISBN 978-970-15-1430-6.